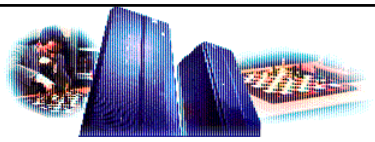
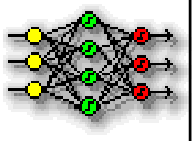




Sistemi Intelligenti
Reinforcement Learning:
Iterative policy evaluation

Alberto Borghese

Università degli Studi di Milano
Laboratorio di Sistemi Intelligenti Applicati (AIS-Lab)
Dipartimento di Scienze dell'Informazione
borghese@dsi.unimi.it

Sommario

Dalle equazioni di Bellman alla iterative policy evaluation.

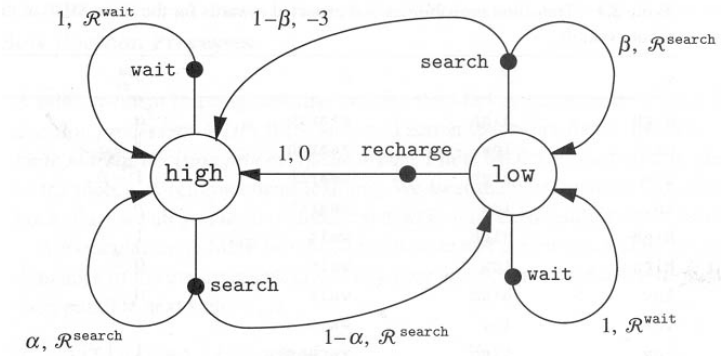
Determinazione della policy ottima.

Osservazioni.

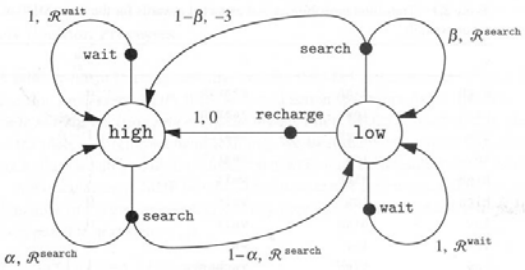
Esempio di calcolo della Value function

Value function
V(high) = ?
V(low) = ?

Policy
a(s=high) = ?
a(s=low) = ?



Esempio di calcolo della funzione valore



α=0.4, β=0.1, γ=0.8, R^{search}=3, R^{wait}=1

$Vh = Pr(W)x1x[1+0.8Vh] + Pr(S)x0.4x[3+0.8xVh]+Pr(S)x0.6x[3+0.8Vl]$

Wait *Search -> high* *Search -> low*

$Vl = Pr(W)x1x[1+0.8Vl] + Pr(S)x0.1x[3+0.8xVl]+Pr(S)x0.9x[-3+0.8Vh]+ Pr(R)x1x[0+0.8Vh]$

Wait *Search -> High* *Recharge* *Search -> Low*

Sistema lineare di 2 equazioni nelle 2 incognite: Vh e Vl
Come calcolo Vh e Vl?

Problematiche legate al calcolo di V(s): problema di policy evaluation

3 assunzioni:

1) Conoscenza della dinamica dell'ambiente: P(s->s'| a_j)
2) Conoscenza della policy (eventualmente stocastica), $\pi(s, a)$
3) Potenza di calcolo sufficiente
4) Proprietà Markoviane dell'ambiente (definizione di uno stato).

Le equazioni contengono dei termini statistici (valori attesi).

Soluzione di un sistema lineare in N incognite (numero di stati).

Come mai posso determinare la Value function per la policy p(), se questa si basa sul reward che riceverò negli istanti futuri?

C'e' poca interazione con l'ambiente e molta simulazione (cf. metodi Montecarlo).

Calcolo ricorsivo della Value function

$$V^{\pi}(s) = E_{\pi}\{R_t \mid s_t = s\} = E_{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s\right\}$$

$$V^{\pi}(s') = E_{\pi}\{R_{t+1} \mid s_{t+1} = s'\}$$

Legame?

Policy Next-state

$$P_{s->s'|a} = \Pr\{s_{t+1} = s' \mid s_t = s, a_t = a\}$$

$$V^{\pi}(s) = \sum_{a_j} \pi(a_j, s) \sum_{s'} P_{s \to s'|a_j} R_{s \to s'|a_j} + E_{\pi}\left\{\gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, s_t = s\right\}$$

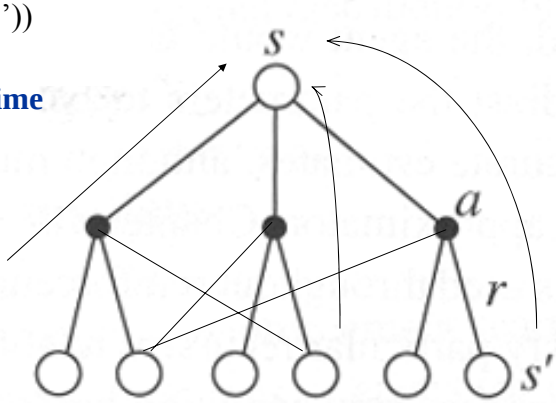
$$V^{\pi}(s) = \left\{\sum_{a_j} \pi(a_j, s) \sum_{s_l'} \left\{P_{s \to s_l'|a_j}\left[R_{s \to s_l'|a_j} + \gamma V^{\pi}(s_l')\right]\right\}\right\}$$

Bellman's equation

Osservazioni

$V^{\pi}(s)$ = funz($V^{\pi}(s')$)

Backwards in time



$$V^{\pi}(s)=\left\{\sum_{a_j}\pi(a_j,s)\sum_{s_l'}\left\{P_{s\to s_l'|a_j}\left[R_{s\to s_l'|a_j}+\gamma V^{\pi}(s_l')\right]\right\}\right\}$$

Iterative policy evaluation

Evoluzione del sistema da s(t=0) a {s'(t = T)} utilizzando la policy π(s,a), prefissata.

Quanto valgono gli stati?

Parto da $V(s(t=0))_{k=0}$ arbitraria, otterrò una value function per ogni stato che sarà funzione di V(s(t=0)).

Devo migliorare, come?

Utilizziamo l'informazione sul **passato**.

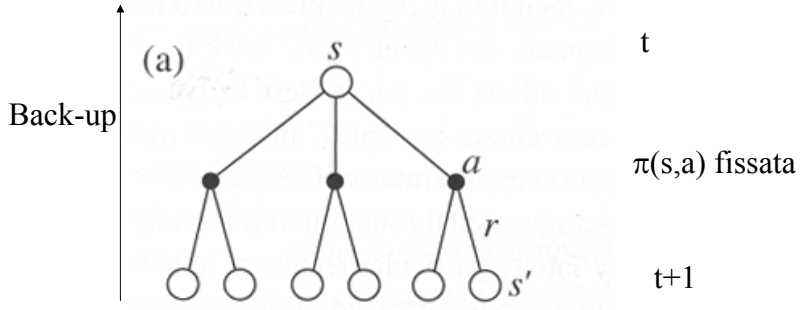
Fondamenti del metodo

• Supponiamo di essere all'istante t. In questo istante t, si può passare ad un certo insieme di stati: $\{s'_{t+1}\}$.

•Analizziamo un solo passo: cosa succede nella transizione da t a t+1.

• Migliorare la stima della nostra Value Function ad ogni iterazione.

Tecnica full-backup



Conosciamo $V_k(s(t))$ $\forall s$, anche per s' quindi
Analizziamo la transizione da s(t) -> {s'(t+1)}
Calcoliamo un nuovo valore di s: $V_{k+1}(s(t))$ congruente con $V_k(s(t))$ ed r_{t+1}
Full backup se esaminiamo tutti gli s' (cf. DP).

Da s' mi guardo indietro ed aggiorno V(s).

Calcolo iterativo della Value Function

Per ogni stato s, estratto a caso, analizziamo una singola transizione.

Equazione di Bellman per "iterative policy evaluation":

$$V_{k+1}(s) = \left[\sum_{a_j} \pi(a_j, s)\right] \sum_{s'} P_{s \to s' | a_j} \left[R_{s \to s' | a_j} + \gamma V_k(s')\right]$$

Mi fido di $V_k(s')$ (Backup)

$$\lim_{k \to \infty} \{V_k(s)\} = V^{\pi}(s)$$

Algoritmo per "iterative policy evaluation"

Partiamo da una politica π(s,a) data.
Inizializziamo V(s) = 0 ∀s, compreso gli stati finali.
Repeat
{ Δ = 0;
for s = 1 : N // ∀s, ≠ TS
{ Value = V(s);

$$V_{k+1}(s) = \sum_{a_j} \pi(s, a_j) \sum_{s'} P^{a_j}_{s->s'} \left[R^{a_j}_{s->s'} + \gamma V(s')\right]$$

Δ = max(Δ, | Value – $V_{k+1}(s)$ |)
}
} Until (Δ < threshold);

Esempio

•L'agente evolve esplorando gli stati 1-14. TS sono gli stati terminali.

•L'agente può spostarsi: {dx, sx, alto, basso}

• $\pi(s,a)$ è equiprobabile.

• R = -1 per ogni transizione, tranne quando s' = TS (R = 0).

• $\gamma = 1$.

TS	1	2	3
4	5	6	7
8	9	10	11
12	13	14	TS

Esempio – $V(s)_{k=0}$

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0


Esempio – $V(s)_{k=1}$

0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

$V(s)_{k=0}$

0	-0.75	-1	-1
-0.75	-1	-1	-1
-1	-1	-1	-0.75
-1	-1	-0.75	0

$V(s)_{k=1}$

Esempi:

- $V(s=10)_{k=1} = 0.25\ [1\ (-1 + 1\ V(14)_{k=0}] + 0.25\ [1\ (-1 + 1\ V(6)_{k=0}] +$
  $0.25\ [1\ (-1 + 1\ V(9)_{k=0}] + 0.25\ [1\ (-1 + 1\ V(11)_{k=0}] = -1$
- $V(s=1)_{k=1} = 0.25\ [1\ (-1 + 1\ V(5)_{k=0}] + 0.25\ [1\ (-1 + 1\ V(5)_{k=0}] +$
  $0.25\ [1\ (0 + 1\ V(TS)_{k=0}] + 0.25\ [1\ (-1 + 1\ V(2)_{k=0}] = -0.75$


Esempio – $V(s)_{k=2}$

0	-0.75	-1	-1
-0.75	-1	-1	-1
-1	-1	-1	-0.75
-1	-1	-0.75	0

$V(s)_{k=1}$

0	-0.75	-1.9375	-2
-0.75	-1.875	-2	-1.9375
-1.9375	-2	-1.875	-0.75
-2	-1.9375	-0.75	0

$V(s)_{k=2}$

Esempi:

- $V(s=10)_{k=1} = 0.25\ [1\ (-1 + 1\ V(14)_{k=1}] + 0.25\ [1\ (-1 + 1\ V(6)_{k=1}] +$
  $0.25\ [1\ (-1 + 1\ V(9)_{k=1}] + 0.25\ [1\ (-1 + 1\ V(11)_{k=0}] = -1$
- $V(s=1)_{k=1} = 0.25\ [1\ (-1 + 1\ V(5)_{k=0}] + 0.25\ [1\ (-1 + 1\ V(5)_{k=0}] +$
  $0.25\ [1\ (0 + 1\ V(TS)_{k=0}] + 0.25\ [1\ (-1 + 1\ V(2)_{k=0}] = -0.75$

Interpretazione dell'update (batch o trial)

$$V_{new}(s) = \sum_{a_j} \pi(s,a_j) \sum_{s'} P^{a_j}_{s->s'} \left[R^{a_j}_{s->s'} + \gamma V(s') \right]$$

Al termine dell'aggiornamento dei V(s) per tutti gli stati, V(s) = $V_{new}(s)$. **Aggiornamento batch**.

$$V(s) = \sum_{a_j} \pi(s,a_j) \sum_{s'} P^{a_j}_{s->s'} \left[R^{a_j}_{s->s'} + \gamma V(s') \right]$$

Utilizzerò in parte già il nuovo valore di V(s) all'interno dell'equazione di aggiornamento. **Aggiornamento per trial.**

Entrambe le modalità di aggiornamento convergono.

Sommario

Dalle equazioni di Bellman alla iterative policy evaluation.

Determinazione della policy ottima.

Osservazioni.

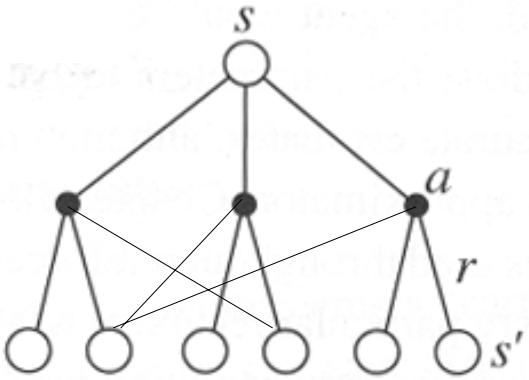
Gli attori nel caso di politica ottima

V*(s)

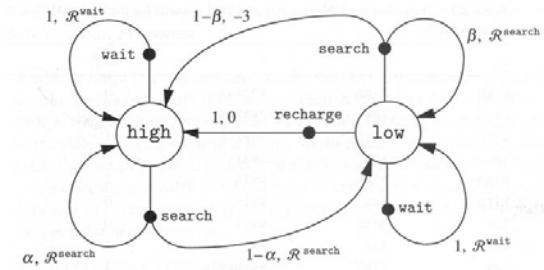
a : argmax(V(s))

Q*(s,a)

V*(s')



Esempio di calcolo di V*(s) - I



$\alpha = 0.4$, $\beta = 0.1$, $\gamma = 0.8$, $R^{Wait} = 1$, $R^{Search} = 3$

V(high):

$$state = high, action = wait \rightarrow Q(high, wait) = 1[R_{wait} + \gamma V^*(high)]$$
$$state = high, action = search \rightarrow Q(high, search) = \alpha[R^{search} + \gamma V^*(high)] + (1-\alpha)[R^{search} + \gamma V^*(low)]$$

V(low):

$$state = low, action = wait \rightarrow Q(low, wait) = [R_{wait} + \gamma V^*(low)]$$
$$state = low, action = search \rightarrow Q(low, search) = \beta[R^{search} + \gamma V^*(low)] + (1-\beta)[-3 + \gamma V^*(high)]$$
$$state = low, action = rechar \rightarrow Q(low, rechar) = \gamma V^*(h)$$

Esempio di calcolo di V*(s) - II

s = HighEnergy	a = Search;	Vh ≈ 6 + 0.65V*l
	a = Wait;	Vh = 5 (per ogni V*l)
s = Low	a = Search;	Vl ≈ -2.2 + 6.6 V*h
	a = Wait;	Vl = 5
	a = Recharge;	Vl = 0.8 V*h

Come calcolo V*h e V*l?

Esempio di calcolo di V*(s) - III

s = HighEnergy	a = Search;	Vh ≈ 6 + 0.65V*l
	a = Wait;	Vh = 5 (per ogni V*l)
s = Low	a = Search;	Vl ≈ -2.2 + 6.6 V*h
	a = Wait;	Vl = 5
	a = Recharge;	Vl = 0.8 V*h

Scelgo $\underset{a}{Max}$ sia per lo stato High Energy che Low Energy

Sistema non-lineare..

Esempio di calcolo di V*(s) - IV

s = HighEnergy	a = Search;	$Vh \approx 6 + 0.65Vl \approx 9.25$
s = Low	a = Wait;	$Vl = 5$
s = HighEnergy	a = Search;	$Vh \approx 6 + 0.65Vl \approx 12$
s = Low	a = Recharge;	$Vl = 0.8 * Vh = 9.6$
s = HighEnergy	a = Wait;	$Vh \approx 5$
s = Low	a = Search;	$Vl = -2.2 + 6.6Vh \approx 33$

Serve comunque una valutazione globale di V(s) per i vari stati.

La soluzione dipende dal valore dei parametri.

Ottimizzazione policy

Per ogni stato scelgo le azioni secondo la policy: $\pi(s,a)$.

Posso ordinare la Value function V(s) in funzione delle azioni scelte in s (policy).

Si definisce una policy, π_1, migliore di un'altra, π_2, se e solo se:
$V^{\pi 1}(s) >= V^{\pi 2}(s)\ \forall s$.

In particolare si definisce una politica ottima, π^*, se e solo se:
$V^*(s) >= V^{\pi}(s)\ \forall s$

$Q^*(s,a) >= Q^{\pi}(s,a)\ \forall [s,a]$

Relazione soddisfatta da V*(s)

$$V^*(s) = \max_a [E_\pi \{R_t \mid s_t = s, a_t\}] =$$

$$\max_a \left[E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \,\middle|\, s_t = s, a_t = a \right\} \right] =$$

$$\max_a \left[r_{t+1} + \gamma E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} \,\middle|\, s_t = s, a_t = a \right\} \right] =$$

$$\max_a [r_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a] =>$$

$$V^*(s) = \max_a \{P_{s->s'|a} [R_{s->s'|a} + \gamma V^*(s')]\}$$

Bellman's Equation For optimal policy

Sommario

Dalle equazioni di Bellman alla iterative policy evaluation.

Determinazione della policy ottima.

Osservazioni.

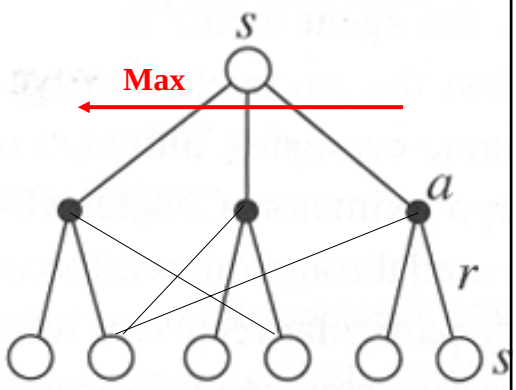
V*(s) – Osservazioni

$$V^*(s) = \max_{a_j} \sum_{s'} P_{s \to s'|a_j} \left[R_{s \to s'|a_j} + \gamma V^*(s') \right]$$

Per ogni stato devo valutare:
•L'azione migliore ad un passo

Come valuto?
• analizzando reward a lungo termine

t



Calcolo ricorsivo della Value function ottima::confronti

$$V^{\pi}(s) = \left\{ \sum_{a_j} \pi(a_j, s) \sum_{s_l'} \left\{ P_{s \to s_l'|a_j} \left[R_{s \to s_l'|a_j} + \gamma V^{\pi}(s_l') \right] \right\} \right\}$$

V*(s) di uno stato, quando viene scelta la policy ottima,
deve essere uguale al valore atteso del reward per l'azione
migliore per lo stato s.

$$V^*(s) = \max_{a_j} \sum_{s'} P_{s \to s'|a_j} \left[R_{s \to s'|a_j} + \gamma V^*(s') \right]$$

Politica greedy: scelgo l'azione ottimale.
Ha senso per il robot raccogli-lattine?

Utilizzo di V(s) per determinare la policy ottima

Esplorazione dell'effetto a lungo termine di tutte le azioni possibili.

Politica greedy rispetto alla Value function.

Questa politica greedy (ad un passo) produce una politica ottima globalmente.

Vengono valutate le conseguente a breve termine delle azioni (1-step) ma non è una politca miope perché consente di ottenere una politica globalmente ottima.

E' reso possibile per la conoscenza dell'ambiente (stocastico).

Problematiche legate al calcolo di V*(s)

Soluzione vicina alla ricerca esaustiva. Devo valutare per ogni stato tutte le possibili azioni (devo trovare il massimo).

Per tutte le possibili azioni devo calcolare la probabilità di transizione allo stato successivo e di ottenere una certa reward.

3 assunzioni:

1) Conoscenza della dinamica dell'ambiente: P(s->s'| a_j)
2) Potenza di calcolo sufficiente
3) Proprietà Markoviane dell'ambiente (definizione di uno stato).

Soluzioni approssimate.

Sommario

Dalle equazioni di Bellman alla iterative policy evaluation.

Determinazione della policy ottima.

Osservazioni.